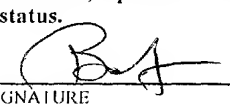


FORM PTO-1390 (REV. 11-2000)		U.S. DEPARTMENT OF COMMERCE PATENT AND TRADEMARK OFFICE	ATTORNEY'S DOCKET NUMBER GIC-560
TRANSMITTAL LETTER TO THE UNITED STATES DESIGNATED/ELECTED OFFICE (DO/EO/US) CONCERNING A FILING UNDER 35 U.S.C. 371			U.S. APPLICATION NO. (If known, see 37 CFR 1.5) 09/831073
INTERNATIONAL APPLICATION NO. PCT/US99/23346	INTERNATIONAL FILING DATE 07 October 1999	PRIORITY DATE CLAIMED 12 November 1998	
TITLE OF INVENTION DIGITAL TELEVISION RECEIVER WITH APPLICATION PROGRAMMING INTERFACE FOR USER MANAGEMENT			
APPLICANT(S) FOR DO/EO/US Petr PETERKA and Branislav N. MEANDZIJA			
Applicant herewith submits to the United States Designated/Elected Office (DO/EO/US) the following items and other information:			
<p>1. <input checked="" type="checkbox"/> This is a FIRST submission of items concerning a filing under 35 U.S.C. 371.</p> <p>2. <input type="checkbox"/> This is a SECOND or SUBSEQUENT submission of items concerning a filing under 35 U.S.C. 371.</p> <p>3. <input checked="" type="checkbox"/> This is an express request to begin national examination procedures (35 U.S.C. 371(f)). The submission must include items (5), (6), (9) and (21) indicated below.</p> <p>4. <input checked="" type="checkbox"/> The US has been elected by the expiration of 19 months from the priority date (Article 31).</p> <p>5. <input checked="" type="checkbox"/> A copy of the International Application as filed (35 U.S.C. 371(c)(2))</p> <p style="margin-left: 20px;">a. <input type="checkbox"/> is attached hereto (required only if not communicated by the International Bureau).</p> <p style="margin-left: 20px;">b. <input type="checkbox"/> has been communicated by the International Bureau.</p> <p style="margin-left: 20px;">c. <input checked="" type="checkbox"/> is not required, as the application was filed in the United States Receiving Office (RO/US).</p> <p>6. <input type="checkbox"/> An English language translation of the International Application as filed (35 U.S.C. 371(c)(2)).</p> <p style="margin-left: 20px;">a. <input type="checkbox"/> is attached hereto.</p> <p style="margin-left: 20px;">b. <input type="checkbox"/> has been previously submitted under 35 U.S.C. 154(d)(4).</p> <p>7. <input checked="" type="checkbox"/> Amendments to the claims of the International Application under PCT Article 19 (35 U.S.C. 371(c)(3))</p> <p style="margin-left: 20px;">a. <input type="checkbox"/> are attached hereto (required only if not communicated by the International Bureau).</p> <p style="margin-left: 20px;">b. <input type="checkbox"/> have been communicated by the International Bureau.</p> <p style="margin-left: 20px;">c. <input type="checkbox"/> have not been made; however, the time limit for making such amendments has NOT expired.</p> <p style="margin-left: 20px;">d. <input checked="" type="checkbox"/> have not been made and will not be made.</p> <p>8. <input type="checkbox"/> An English language translation of the amendments to the claims under PCT Article 19 (35 U.S.C. 371(c)(3)).</p> <p>9. <input checked="" type="checkbox"/> An oath or declaration of the inventor(s) (35 U.S.C. 371(c)(4)).</p> <p>10. <input type="checkbox"/> An English language translation of the annexes of the International Preliminary Examination Report under PCT Article 36 (35 U.S.C. 371(c)(5)).</p> <p>Items 11 to 20 below concern document(s) or information included:</p> <p>11. <input type="checkbox"/> An Information Disclosure Statement under 37 CFR 1.97 and 1.98.</p> <p>12. <input checked="" type="checkbox"/> An assignment document for recording. A separate cover sheet in compliance with 37 CFR 3.28 and 3.31 is included.</p> <p>13. <input type="checkbox"/> A FIRST preliminary amendment.</p> <p>14. <input type="checkbox"/> A SECOND or SUBSEQUENT preliminary amendment.</p> <p>15. <input type="checkbox"/> A substitute specification.</p> <p>16. <input type="checkbox"/> A change of power of attorney and/or address letter.</p> <p>17. <input type="checkbox"/> A computer-readable form of the sequence listing in accordance with PCT Rule 13ter.2 and 35 U.S.C. 1.821 - 1.825.</p> <p>18. <input type="checkbox"/> A second copy of the published international application under 35 U.S.C. 154(d)(4).</p> <p>19. <input type="checkbox"/> A second copy of the English language translation of the international application under 35 U.S.C. 154(d)(4).</p> <p>20. <input checked="" type="checkbox"/> Other items or information:</p> <p style="margin-left: 20px;">a) Patent Application Data Entry Form - 1 sheet</p> <p style="margin-left: 20px;">b) Patent Application Specification, including claims and Abstract - 30 sheets (copy for US examination - international application as amended on March 22, 2001)</p>			

- c) Five (5) sheets of formal drawings, together with separate transmittal letter
- d) Express Mail Certificate

U.S. APPLICATION NO. 09/831073 INTERNATIONAL APPLICATION NO. PCT/US99/23346		ATTORNEY'S DOCKET NUMBER GIC-560	
21. <input checked="" type="checkbox"/> The following fees are submitted: BASIC NATIONAL FEE (37 CFR 1.492 (a) (1) - (5)): Neither international preliminary examination fee (37 CFR 1.482) nor international search fee (37 CFR 1.445(a)(2)) paid to USPTO and International Search Report not prepared by the EPO or JPO. \$1000.00 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but International Search Report prepared by the EPO or JPO \$860.00 International preliminary examination fee (37 CFR 1.482) not paid to USPTO but international search fee (37 CFR 1.445(a)(2)) paid to USPTO \$710.00 International preliminary examination fee (37 CFR 1.482) paid to USPTO but all claims did not satisfy provisions of PCT Article 33(1)-(4) \$690.00 International preliminary examination fee (37 CFR 1.482) paid to USPTO and all claims satisfied provisions of PCT Article 33(1)-(4) \$100.00 ENTER APPROPRIATE BASIC FEE AMOUNT =		CALCULATIONS PTO USE ONLY	
Surcharge of \$130.00 for furnishing the oath or declaration later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492(e)).		\$ 690.00 \$ ---	
CLAIMS	NUMBER FILED	NUMBER EXTRA	RATE
Total claims	18 - 20 =	0	x \$18.00
Independent claims	5 - 3 =	2	x \$80.00
MULTIPLE DEPENDENT CLAIM(S) (if applicable)			+ \$270.00
TOTAL OF ABOVE CALCULATIONS =		\$ 850.00	
<input type="checkbox"/> Applicant claims small entity status. See 37 CFR 1.27. The fees indicated above are reduced by 1/2.		+	
SUBTOTAL =		\$ 850.00	
Processing fee of \$130.00 for furnishing the English translation later than <input type="checkbox"/> 20 <input type="checkbox"/> 30 months from the earliest claimed priority date (37 CFR 1.492(f)).		\$ ---	
TOTAL NATIONAL FEE =		\$ 850.00	
Fee for recording the enclosed assignment (37 CFR 1.21(h)). The assignment must be accompanied by an appropriate cover sheet (37 CFR 3.28, 3.31). \$40.00 per property +		\$ 40.00	
TOTAL FEES ENCLOSED =		\$ 890.00	
		Amount to be refunded:	\$
		charged:	\$
a. <input checked="" type="checkbox"/> A check in the amount of \$ <u>890.00</u> to cover the above fees is enclosed. b. <input type="checkbox"/> Please charge my Deposit Account No. _____ in the amount of \$ _____ to cover the above fees. A duplicate copy of this sheet is enclosed. c. <input checked="" type="checkbox"/> The Commissioner is hereby authorized to charge any additional fees which may be required, or credit any overpayment to Deposit Account No. <u>50-0625</u> . A duplicate copy of this sheet is enclosed. d. <input type="checkbox"/> Fees are to be charged to a credit card. WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.			
NOTE: Where an appropriate time limit under 37 CFR 1.494 or 1.495 has not been met, a petition to revive (37 CFR 1.137 (a) or (b)) must be filed and granted to restore the application to pending status.			
SEND ALL CORRESPONDENCE TO Barry R. Lipsitz Law Offices of Barry R. Lipsitz 755 Main Street, Building No. 8 Monroe, CT 06468 (203) 459-0200 Date: <u>03 May 2001</u>			
		 SIGNATURE Barry R. Lipsitz NAME <u>28,637</u> REGISTRATION NUMBER	Express Mail No.: <u>EL 827 616 075 US</u>

5/PRTS

JC08 Rec'd PCT/PTO 03 MAY 2001

1

DIGITAL TELEVISION RECEIVER WITH APPLICATION
PROGRAMMING INTERFACE FOR USER MANAGEMENT

BACKGROUND OF THE INVENTION

5 This application claims the benefit of U.S.
Provisional Application No. 60/107,949, filed November
12, 1998.

10 The present invention provides an Application
Programming Interface (API) for downloadable broadcast
applications to manage/access user-related information
on a Digital Television (DTV) Receiver/Terminal.

A set-top terminal, also referred to as an
Integrated Receiver-Decoder (IRD) or a subscriber
terminal, is a device that receives and decodes
television signals for presentation by a television.
15 The signals can be delivered over a satellite, through
a cable plant, or by means of terrestrial broadcast,
for example. Various applications have been proposed,
or are currently available, via modern set tops,
including video on demand (VOD), audio on demand, pay-
20 per-view, interactive shopping, electronic commerce,
electronic program guides, Internet browsers, mail
services (e.g., text e-mail, voice mail, audio mail,
and/or video mail), telephony services, stock ticker,
weather data, travel information, games, gambling,
25 banking, shopping, voting, and others. Applications
may also enable Internet connectivity and possibly

Internet-based telephony. The set top functionality is enabled through specialized hardware and software.

5 The applications may be downloaded by terminals via a network, loaded locally (e.g., via a smart card), or installed at the time of manufacture, for example.

10 Moreover, with the increasing integration of computer networks such as the Internet, telephony networks, and broadband distribution networks, many opportunities arise for providing new types of applications.

15 To optimize the user's ability to access these applications, there is a need for user management at the terminals. In particular, it would be desirable to provide a user-management API that can be implemented in software independently of a terminal's hardware and operating system.

20 The API should allow new users to be recognized, or old users to be deleted, as well as recognizing a current user of the terminal. The API should provide a profile of each user and maintain a record of permissions that have been granted to the users, e.g., for accessing applications such as pay-per-view, or enforcing rating guidelines for children.

25 The API should maintain a record of user preferences, such as preferred language and rating ceilings, personal data such as age, address, zip code, account numbers, as well as system-wide preferences, such as favorite channels, language, etc.

30 The API should also maintain a registry of applications, resources, preferences and users. An application may use/access a resource, which is usually

a device, function or a process on the receiver (e.g. tuner, modem, database, etc.)

5 The API should be compatible with Java(tm),
ActiveX(tm) or an equivalent type of component based
object-oriented technology.

10 The API should be compatible with Digital Audio
Visual Council (DAVIC), American Television Standards
Committee (ATSC) T3/S17 Digital TV Application Software
Environment (DASE), Digital Video Broadcast (DVB)-
Multi-Media Home Platform (MHP) and other related
environments.

15 The API should be compatible with any application
at a terminal, regardless of how the application was
received or installed (e.g., downloaded, resident, from
smart card, etc.)

 The present invention provides a system having the
above and other advantages.

SUMMARY OF THE INVENTION

The present invention provides a software architecture a set-top television terminal. In particular, an Application Programming Interface (API) is provided for applications to manage/access user-related information on the terminal.

A "user" is one who is watching the TV or using its other functions, e.g., E-mail, games, etc.

The API includes a user software package that includes registry, profile, permissions, change cause, change event, and registry event functions. A preferences package includes a registry, names, rating, preferred language, change cause, change event and registry event functions. A registry package includes type, factory, change event, listener, user, preference, resource and application functions.

The present invention provides a multi-user environment (user registry and user profile), and a registry for preferences which can be associated with an individual user (user-specific) or be common to all users (system-wide). A set of permissions is associated with each user, or at least with a default user, which represents anybody watching the TV.

In a particular embodiment, a television set-top terminal includes a computer readable medium having computer program code means, and means for executing the computer program code means to implement an Application Programming Interface (API). The API provides (a) a user registry of a plurality of users of the terminal, (b) a preferences registry of preferences

of the users, and (c) permission(s) for controlling the users' access to at least one application that is provided at the terminal.

5 The API provides a security policy to allow only specified applications to access the preferences registry. The security policy may be user-controlled. Thus, some applications can, and some cannot, access the preferences based on a security policy. This means that the user may allow only specified trusted
10 applications to access his/her credit card number but not others, for example. These applications may be specified shop-at-home channels or the like.

The API disallows access to user preferences by an application that does not have the required
15 permission(s).

The permission(s) may be associated with each user individually, or may be associated with a default user. A default user might be provided to indicate that the entire family is watching the television together, or
20 to indicate a guest or unknown person.

Additionally, the application generally is responsive to the user preferences. For example, if an application such as an Electronic Program Guide (EPG) accesses a language preference of a particular user,
25 the application can automatically select the language for text when the service information (SI) is multilingual. An application that tunes to channels (audio/video or audio-only) may automatically select the appropriate audio language based on the user's
30 language preference.

As another example, an EPG application may access

favorite channel information of a user to prepare a custom made guide to the programs that are currently playing. Or, an application which enables targeted advertisements may access information regarding a user's location (ZIP code), gender, age, family status, and other personal information, such as pets, hobbies, etc.

Additionally, e-commerce-type applications may need to look at user credit card numbers, permissions to do on-line purchases, and other related preferences.

The permission may also provide, for example, a parental control/rating ceiling or permission to buy Impulse Pay Per View (IPPV) programs, or engage in e-commerce transactions (e.g., purchase goods or services via the television - home shopping).

The preferences include a ratings ceiling preference.

Moreover, the user preferences may include system-wide or user-specific.

The user registry can be used to register a new user of the terminal, identify a current user of the terminal, and remove a former user of the terminal.

For example, when an application needs to switch between users, it may do it by identifying the user using a logon screen, by PIN codes or passwords, or even personalized smart cards, or voice control, for example.

The permission(s) enable the users to access a resource/invoke functions of the terminal. This refers to accessing either some physical resource on the receiver, such as a modem or a tuner, the smart card,

etc., or an application, such as e-mail, web browsing, e-commerce, etc.

5 The resource may be, for example, a device, function or a process on the receiver, such as a tuner, modem, database, plug-in module, cable, software module, network interface card, persistent storage, TV screen space, memory, CPU, conditional access (CA) module, and so forth.

10 The API also is adapted to define one or more new types of user preferences, and add the new types of user preferences to the preferences registry. The new user preferences can be defined by a base interface, Preference, which is an object that can be added or removed from the PreferenceRegistry. This allows one
15 to define new types of user preferences, e.g., by extending it the same way as the LanguagePreference extends the base Preference.

For example, the preferences may include specific types of programs (sports, drama, humor, action, etc.).

20 The API also is adapted to associate the new types of user preferences with the users.

The API may be independent of an operating system and hardware of the terminal.

A corresponding method is also presented.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows package relationships and dependencies in accordance with the present invention.

5 FIG. 2 illustrates a user class/interface diagram in accordance with the present invention.

FIG. 3 illustrates a preferences class/interface diagram in accordance with the present invention.

FIG. 4 illustrates a registry class/interface diagram in accordance with the present invention.

DETAILED DESCRIPTION OF THE INVENTION

1. INTRODUCTION

The present invention provides an Application Programming Interface (API) for downloadable broadcast applications to manage/access user-related information on a Digital Television (DTV) Receiver.

The invention comprises the following two API components: 1) User API; and 2) User Preferences API.

They can be used together where each user has a set of user preferences associated with it, or separately where there is a system-level set of preferences without a user association.

A Registry package is also described herein, which provides a generic mechanism for registry-type constructs.

2. MODEL DESCRIPTION

FIG. 1 illustrates an API package in accordance with the present invention, namely a User package 110, a Preferences package 120, and a Registry package 130. These packages are described in the following sections, in connection with FIGs 2-4.

Note that portions of the disclosure were generated automatically from Rational Rose(tm) CASE tool, developed by Rational Software Corporation, USA. The figures use the Rational Rose (tm) depiction of the Unified Modeling Language (UML), which is a language for specifying, constructing, visualizing, and documenting the artifacts of a software-intensive system. A class diagram represents the static

structure of a system, and shows a pattern of behaviors that the system exhibits. This is accomplished by showing the existence of classes and their relationships. Each class is represented by a box with three sections. The top section lists the class name. The middle section denotes a list of attributes, and the bottom section denotes a list of operations.

A solid or dashed line between classes denotes an association or dependency. A white diamond tip denotes aggregation by reference, while a black diamond tip denotes aggregation by value. A triangular arrowhead denotes a restricted navigation, e.g., inheritance of operation but not of structure.

Moreover, interfaces and classes begin with an uppercase letter, while methods begin with a lowercase letter. A class is a template that defines a data structure, method and function calls for an object. An interface defines a set of methods/function calls that can be manipulated by a class. The class provides the code for implementing an interface.

2.1 User Package

FIG. 2 illustrates a user class/interface diagram in accordance with the present invention. A main UserRegistry interface 210 is the access point for getting information about any user. One or more users can be associated with each terminal (e.g., DTV receiver, set-top box, IRD, TV-enabled PC, etc.) Existing users can be retrieved, new ones can be created, and the current active user can be set via the UserRegistry object 210. The user is represented by the

UserProfile object 220, which holds the user's name and the user's preferences. The preferences may be that the user prefers to view/use certain types of television programs (e.g., favorite channels) and/or applications, preferred audio language, personal data, such age, location (address/zip code), credit card numbers, etc. The user can be authenticated by invoking the implementation-neutral method authenticate(), which may invoke the particular mechanism of authenticating a user, such as Personal Identification Number (PIN) codes, passwords, etc. Users are associated with permissions which are internally used to enforce a security policy. Therefore, new permissions can be granted to each user. The permissions are implementation-specific since they are internal to the terminal.

2.2 Preferences Package

FIG. 3 illustrates a preferences class/interface diagram in accordance with the present invention.

The preference package 120 serves two primary purposes: to hold system-wide preferences and to hold user-specific preferences.

A system-wide preference is anything that applies to all users. For example, English may be a system wide preferred language which applies to all user unless they override it with their own user-specific preference (e.g., Spanish).

When the preference is used in the system-wide sense, the access to it is provided via the PreferenceRegistry 310 obtained from the

RegistryFactory 430 (FIG. 4). The user-specific preferences can be obtained directly from the user (UserProfile 220).

Any new preferences or settings must derive from the abstract Preference interface 330, which provides the listener mechanism and access to the unique preference name. The derived class/interface can provide any methods, behavior and data structures needed to describe the specific preference or settings object.

Each application can register as a listener (using the `addPropertyChangeListener()` method) to a specific preference and be notified via the `propertyChange()` method it must implement.

Properties are uniquely identified by their names. If an application tries to store a Property with a duplicate name, the `PreferenceAlreadyExists` Exception 390 will be thrown. If the application does not have enough privileges to perform any of the PreferenceRegistry 310 methods, the `AccessDeniedException` 385 will be thrown.

`PropertyChangeEvent` 350 is defined in the standard Java packages (JDK 1.2). Also, the notations "from security" in class 385, and "from beans" refer to classes ("Security" and "Beans", respectively) that are defined in JDK 1.2.

2.3 Registry Package

FIG. 4 illustrates a registry class/interface diagram in accordance with the present invention.

The Registry package 130 provides a basic mechanism to construct a Registry object of any kind. The Registry interface 410 is a base interface which is extended by all specific Registries, such as the
5 UserRegistry 210 or the PreferenceRegistry 310.

A RegistryListener interface 440 and a RegistryChangeEvent interface 250 are associated with this package. The listener interface 440 is used by any object that wants to be notified of any changes in
10 the Registry 410. Changes are considered those that affect the Registry 410 itself (not necessarily the individual elements in the registry), such as adding or removing elements to/from the Registry. The RegistryChangeEvent 250 is an abstract class which is
15 extended by the specific registry events.

Since most of the API is defined in terms of Java Interfaces, the RegistryFactory 430 is a class that hides the actual object construction implementation.

The notation "from util" in the EventObject class
20 415 refers to a class "Util" that is defined in JDK 1.2.

The notation "from resource" in the ResourceRegistry interface 480 refers to a class "Resource" that is discussed in commonly-assigned PCT
25 Patent Application No. PCT/US99/23358, filed October 7, 1999, and entitled "Application Programming Interface (API) For Accessing And Managing Resources In A Digital Television Receiver."

The notation "from application" in the ApplicationRegistry interface 490 refers to a class
30 "Application" that is discussed in commonly-assigned

PCT Patent Application No. PCT/US99/23721, filed October 7, 1999, and entitled "Software Application Lifecycle And Management For Broadcast Applications."

3. CLASS AND INTERFACE DESCRIPTION

5 The following sections provide the details of each class, interface and its methods.

3.1 User Package

This package 110 provides classes and interfaces necessary for user management functions.

10 3.1.1 UserRegistry

UserRegistry interface 210 provides access to all users defined on the system. It is derived from Registry 130.

Public Operations:

15 **createUser (name : String) : UserProfile**

This method will create a new user of the specified name.

Public operations are those methods that may be called and used by other objects since they are visible outside of the object (e.g., class). In contrast, private operations are visible only to the class itself.

getCurrentUser () : UserProfile

25 This method returns the user who is currently registered as the active user.

getUserNames () : String[]

This method returns a list of all known users in the registry.

setCurrentUser (newUser : String) : UserProfile

This method defines a new current user. The implementation may prompt the user for a password or some other method of authentication.

5 **getUser (name : String) : UserProfile**

This method returns a UserProfile object of the specified name.

3.1.2 **UserProfile**

10 This interface 220 represents a container of a single user information, such as settings and preferences, billing info, etc. it is derived from UserPermissions '230.

Public Operations:

15 **getName () : String**

Returns the name of this user.

getPreferences () : PreferenceRegistry

 This method returns this user's PreferenceRegistry. All operations performed on the returned list of preferences will be reflected in this
20 UserProfile.

authenticate () : boolean

 This method is called to authenticate a user. It invokes an implementation-specific mechanism for user authentication. It may use a user dialog to ask for a
25 password or PIN, or other authentication mechanisms.

 Returns TRUE if authentication succeeded; FALSE otherwise.

grantPermission (newPermission : String) : void

30 This method is called to add a new permission to the user. See UserPermissions.

3.1.3 UserPermissions

This interface 230 defines a list of user permissions that can be granted to a user.

Public Attributes:

5 **IPPV : String = "IPPV Purchase"**
 RATING : String = "Rating Override"

3.1.4 UserRegistryEvent 260

Derived from RegistryChangeEvent 250.

Public Operations:

10 **getUserName () : java.lang.String**

This method returns the User Name of the User that caused this event.

3.1.5 UserChangeCause

15 This interface 240 defines possible causes for the UserRegistryEvent 260.

Public Attributes:

USER_ADDED : short = 1
 USER_REMOVED : short = 2
 NEW_CURRENT_USER : short = 3

20 3.2 Preferences Package

This package 120 defines a set of interfaces and classes which provide a mechanism to define a set of preferences, either at the system level or at the user level.

25 3.2.1 PreferenceAlreadyExistsException

This exception 390 signals that a preference of the same name is already present in the registry and cannot be inserted (added) again. It is derived from Exception, which is in the JDK 1.2 core Java packages -
 5 java.lang.Exception.

3.2.2 PreferenceRegistry

This interface 310 represents a registry of all settings and preferences that can be shared by multiple applications. It is derived from Registry 130.

10 Public Operations:

getPreference (preferenceName : String) :
org.atssc.preferences.Preference

This method returns the preference of the specified name, or null if the name is unknown to the
 15 registry.

addPreference (aPreference :
org.atssc.preferences.Preference) : void

This method allows an application to insert a new preference object into the registry. The new
 20 preference must be of a unique name.

removePreference (preferenceName : String) : void

This method allows an application to remove a preference object from the registry.

listPreferences () :
 25 **org.atssc.preferences.Preference[]**

This method returns a list of all Preferences currently stored in the registry.

3.2.3 PreferredLanguage

This interface 340 is an example of a language preference interface. An ordered list of ISO 639.2 alpha-3 strings is used. The first language in the list is the most desirable language. It is derived from the Preference interface 330.

Public Operations:

getLanguage () : String[]

This method returns an ordered list (most desirable first) of three letter ISO language codes.

setLanguage (valueList : String[]) : void

This method allows an application to change the language preference. It returns the new list.

3.2.4 RatingPreference

This interface 335 represents a parental rating preference based, e.g., on age. It is derived from the Preference interface 330.

Public Operations:

getRatingCeiling () : String

This method returns the maximum rating value allowed for watching.

setRatingCeiling (aValue : String) : String

This method allows an application to change the rating ceiling. It returns the new rating level.

3.2.5 Preference

This is a top-level interface 330 which is common for all preference/settings subinterfaces. It supports the listener model and provides the preference name.

This interface can be extended to support specific preferences with specific access methods. It is derived from PreferenceNames 320.

Public Operations:

5 **addPropertyChangeListener (aListener :
PropertyChangeListener) : void**

This method allows applications interested in changes to this preference to register for preference change events.

10 **removePropertyChangeListener (aListener :
PropertyChangeListener) : void**

This method allows a preference change listener to remove itself from the list of listeners.

15 **getPreferenceName () : String**

This method returns a unique preference name.

3.2.6 PreferenceNames

This interface 320 contains a list of predefined preference names.

Public Attributes:

20 **SIMPLE_RATING : String = "Simple Rating"**

LANGUAGE : String = "Language"

3.2.7 PreferenceRegistryEvent

25 This event 360 informs the RegistryListener 440 about changes in the PreferenceRegistry 310. It is derived from RegistryChangeEvent 250.

Public Operations:

getPreference () : org.atssc.preferences.Preference

Returns the preference that has changed in the repository. This change concerns the repository, such

as adding or removing a Preference from the PreferenceRegistry 310, not a change in the value of the Preference.

3.2.8 PreferenceChangeCause

5 This interface 345 defines reasons for a PreferenceRegistryEvent 360.

Public Attributes:

PREFERENCE_ADDED : short = 1

PREFERENCE_REMOVED : short = 2

10 3.3 Registry Package

This package 130 provides a set of supporting and utility classes and interfaces used by other packages.

3.3.1 Registry

15 This interface 410 provides a common root to all specialized registry interfaces, such ApplicationRegistry 490, ResourceRegistry 480, PreferenceRegistry 310, etc. It is provided so that the RegistryFactory 430 can return a base type.

20 A "base type" is known from the field of object-oriented programming. To illustrate, one can define a class with a set of functions (methods) and internal variables (e.g., a class "Fruit" which represents fruit and its basic characteristics). One can specialize it by defining a new class, "Apple", which inherits
25 everything from the class "Fruit", and adds new functions that are applicable only to Apples but not to Fruit in general. "Fruit" is then referred to as a "base class" or a "base type."

The Registry interface 410 is derived from RegistryType 405.

Public Operations:

getRegistryType () : String

5 Called to determine the type of registry implemented by the object returned by the RegistryFactory's (430) method getRegistry().

addRegistryListener (listener: RegistryListener) :

10 Called to register for events generated by the Registry 410.

removeRegistryListener (listener : RegistryListener) :

 Called to de-register for events generated by the Registry 410.

15 3.3.2 RegistryFactory

 This class 430 provides a mechanism to create objects that implement specific Registry interfaces, such as the ApplicationRegistry 490. This class is modeled after the Factory Method design pattern, which, as is known from the field of object-oriented programming, is a methodology and structure for solving a problem.

Public Operations:

RegistryFactory () :

25 Constructor

getRegistry (registryName : String) : org.atssc.registry.Registry

30 Returns an instance of an object which implements the specified registry interface. Returns null when specified registry does not exist or cannot be created.

The type of the returned object will be one of the derived Registry types, such as the ApplicationRegistry 490.

3.3.3 RegistryType

5 This interface 405 defines names for different registry types, such as an application registry, etc.

Public Attributes:

APPLICATION_REGISTRY : String = "Application Registry"

10 **RESOURCE_REGISTRY** : String = "Resource Registry"

PREFERENCE_REGISTRY : String = "Preference Registry"

USER_REGISTRY : String = "User Registry"

3.3.4 RegistryListener

15 This interface 440 allows an object to listen to changes made to the Registry 410.

Public Operations:

registryChange () : ApplicationRegistryEvent

20 This method of all registered ApplicationRegistryListeners is called by the ApplicationRegistry object 490 when an ApplicationRegistryEvent is fired (emitted).

3.3.5 RegistryChangeEvent

25 This event 250 is a generic registry change event that is extended by all specific registries (such as ApplicationRegistry 490, etc.) to provide specific information about the change. It is derived from EventObject 415.

Public Operations:

getRegistryType () : java.lang.String

Returns the type of a registry that this event is associated with.

5 **getCause () : short**

Returns the cause of the RegistryChangeEvent 250. Each derived event will define a set of causes appropriate for the registry it represents.

10 Accordingly, it can be seen that the present invention provides an API for applications to manage/access user-related information on a Digital Television (DTV) Receiver/Terminal. The API provides a multi-user environment, and a registry for preferences which can be associated with an individual user or be
15 common to all users. A set of permissions is associated with each user, or at least with a default user. The permission may provide, for example, a parental control/rating ceiling or permission to buy Impulse Pay Per View (IPPV) programs, or engage in e-
20 commerce transactions (e.g., purchase goods or services via the television). The invention also supports a mechanism where some applications can, and some cannot, access the preferences based on a security policy. This means that the user may allow only specified
25 trusted applications to access his/her credit card number but not others, for example, for an e-commerce application.

30 Although the invention has been described in connection with various specific embodiments, those skilled in the art will appreciate that numerous

adaptations and modifications may be made thereto without departing from the spirit and scope of the invention as set forth in the claims.

5 For example, while various syntax elements have been discussed herein, note that they are examples only, and any syntax may be used.

10 Moreover, the invention is suitable for use with virtually any type of network, including cable or satellite television broadband communication networks, local area networks (LANs), metropolitan area networks (MANs), wide area networks (WANs), internets, intranets, and the Internet, or combinations thereof.

Additionally, known computer hardware, firmware and/or software techniques may be used to implement the invention.

What is claimed is:

1. A television set-top terminal, comprising:
a computer readable medium having computer program code means; and

means for executing said computer program code means to implement an Application Programming Interface (API) for accessing user-related information at the terminal, wherein:

the API provides: (a) a user registry of a plurality of users of the terminal, (b) a preferences registry of preferences of the users, and (c) permission(s) for controlling the users' access to at least one application that is provided at the terminal; and

the API provides a security policy to allow only specified applications to access the preferences registry.

2. The terminal of claim 1, wherein:
the security policy is user-controlled.

3. The terminal of claim 1, wherein:
the permission(s) are associated with each user.

4. A television set-top terminal, comprising:
a computer readable medium having computer program code means; and

means for executing said computer program code means to implement an Application Programming Interface

(API) for accessing user-related information at the terminal, wherein:

the API provides: (a) a user registry of a plurality of users of the terminal, (b) a preferences registry of preferences of the users, and (c) permission(s) for controlling the users' access to at least one application that is provided at the terminal; and

the permission(s) are associated with a default user.

5. The terminal of claim 1, wherein:

the at least one application is responsive to the user preferences.

6. The terminal of claim 1, wherein:

the permission(s) control the users' access to a plurality of applications that are provided at the terminal.

7. A television set-top terminal, comprising:

a computer readable medium having computer program code means; and

means for executing said computer program code means to implement an Application Programming Interface (API) for accessing user-related information at the terminal, wherein:

the API provides: (a) a user registry of a plurality of users of the terminal, (b) a preferences registry of preferences of the users, and (c) permission(s) for controlling the users' access to at

least one application that is provided at the terminal;
and

the preferences include a language preference.

8. The terminal of claim 1, wherein:
the preferences include a ratings ceiling
preference.

9. A television set-top terminal, comprising:
a computer readable medium having computer program
code means; and

means for executing said computer program code
means to implement an Application Programming Interface
(API) for accessing user-related information at the
terminal, wherein:

the API provides: (a) a user registry of a
plurality of users of the terminal, (b) a preferences
registry of preferences of the users, and (c)
permission(s) for controlling the users' access to at
least one application that is provided at the terminal;
and

said user preferences include system-wide
preferences.

10. The terminal of claim 9, wherein:
said user preferences include user-specific
preferences that override said system-wide preferences.

11. The terminal of claim 1, wherein:
the user registry registers a new user of the
terminal.

12. The terminal of claim 1, wherein:
the user registry identifies a current user of the terminal.

13. The terminal of claim 1, wherein:
the user registry removes a former user of the terminal.

14. The terminal of claim 1, wherein:
the permission(s) enable the users to access a resource of the terminal.

15. The terminal of claim 1, wherein:
the API is adapted to define at least one new type of user preference, and add the new type of user preference to the preferences registry.

16. The terminal of claim 16, wherein:
the API is adapted to associate the new type of user preference with the users.

17. The terminal of claim 1, wherein:
the API disallows access to user preferences by an application that does not have the required permission(s).

18. A method for implementing a software architecture for a television set-top terminal, comprising the steps of:

providing a computer readable medium having computer program code means; and

executing said computer program code means to implement an Application Programming Interface (API) to provide:

(a) a user registry of a plurality of users of the terminal, (b) a preferences registry of preferences of the users, and (c) permission(s) for controlling the users' access to at least one application that is provided at the terminal wherein:

the API provides a security policy to allow only specified applications to access the preferences registry.

ABSTRACT

An Application Programming Interface (API) for applications to manage/access user-related information on a Digital Television (DTV) Receiver/Terminal. The API provides a multi-user environment (using user registry (210) and user profile (220) classes), and a registry for preferences (310) which can be associated with an individual user (user-specific) or be common to all users (system-wide). A set of permissions (230) is associated with each user, or at least with a default user, which represents anybody watching the TV. The permission may provide, for example, a parental control/rating ceiling or permission to buy Impulse Pay Per View (IPPV) programs, or engage in e-commerce transactions (e.g., purchase goods or services via the television). The invention also supports a mechanism (220) where some applications can, and some cannot, access the preferences based on a security policy. This means that the user may allow only specified trusted applications to access his/her credit card number but not others, for example, for an e-commerce application.

PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of:

P. Peterka, et al.

Filed: Herewith

For: Digital Television Receiver with Application Programming Interface for User Management

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as Express Mail (No. EL 827 616 075 US) in an envelope addressed to: BOX PCT, Commissioner for Patents, Washington, D.C. 20231 on: May 3, 2001

By: Cathy Dunne
Cathy Dunne

DRAWING REVIEW BRANCH

Commissioner for Patents
Washington, D.C. 20231

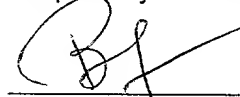
TRANSMITTAL OF FORMAL DRAWING(S)

Dear Sir:

Enclosed are FIVE (5) sheets of formal drawings for filing in the above-referenced patent application.

Please advise the undersigned attorney if correction is necessary.

Respectfully submitted,



Barry R. Lipsitz
Attorney for Applicant(s)
Registration No. 28,637
755 Main Street, Bldg. No. 8
Monroe, CT 06468
(203) 459-0200

Date: May 3, 2001
ATTORNEY DOCKET NO.: GIC-560

1/5

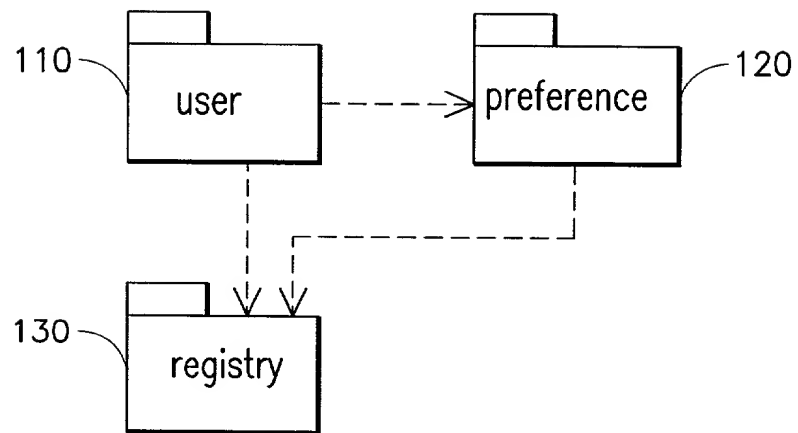


FIG. 1

2/5

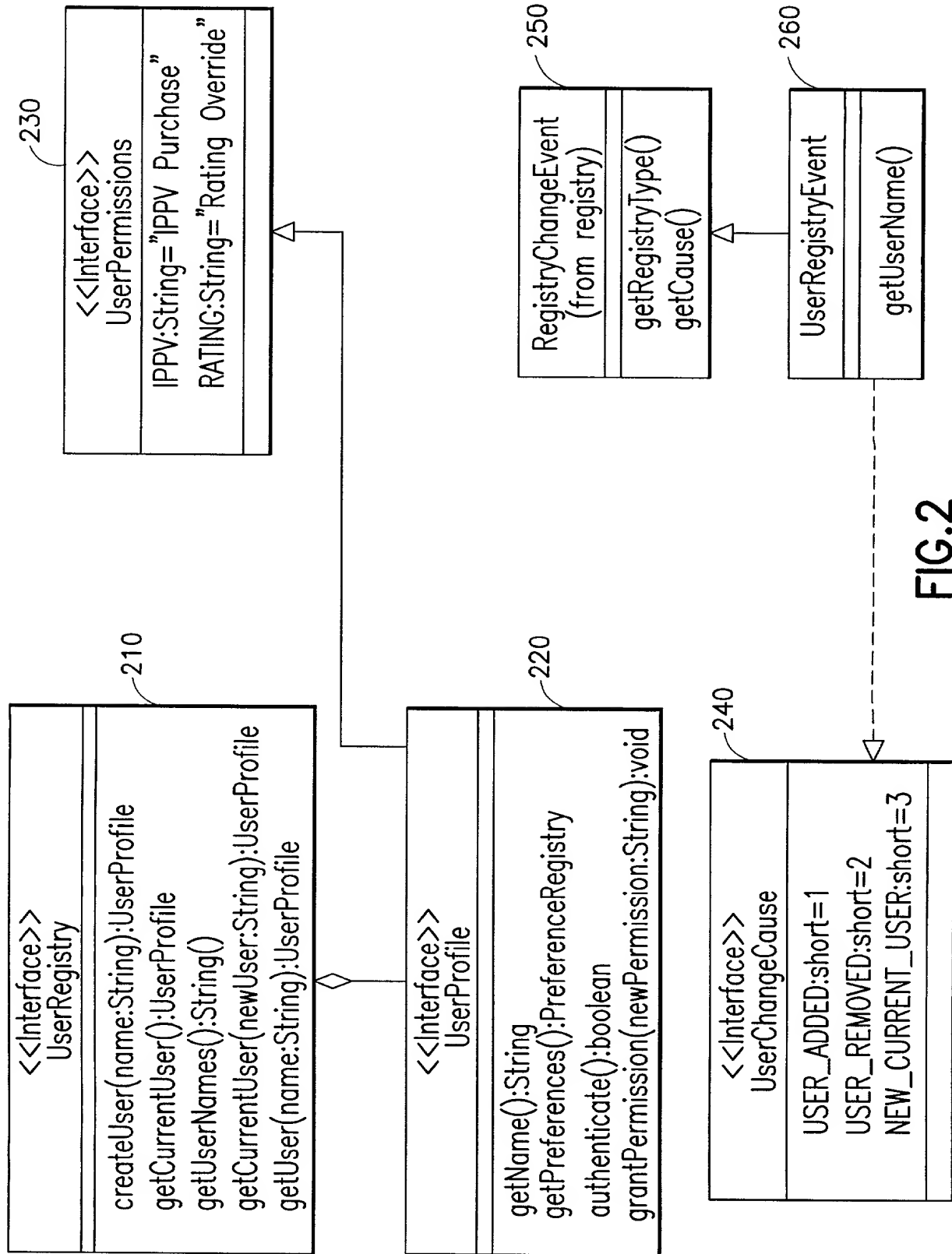


FIG.2

3/5

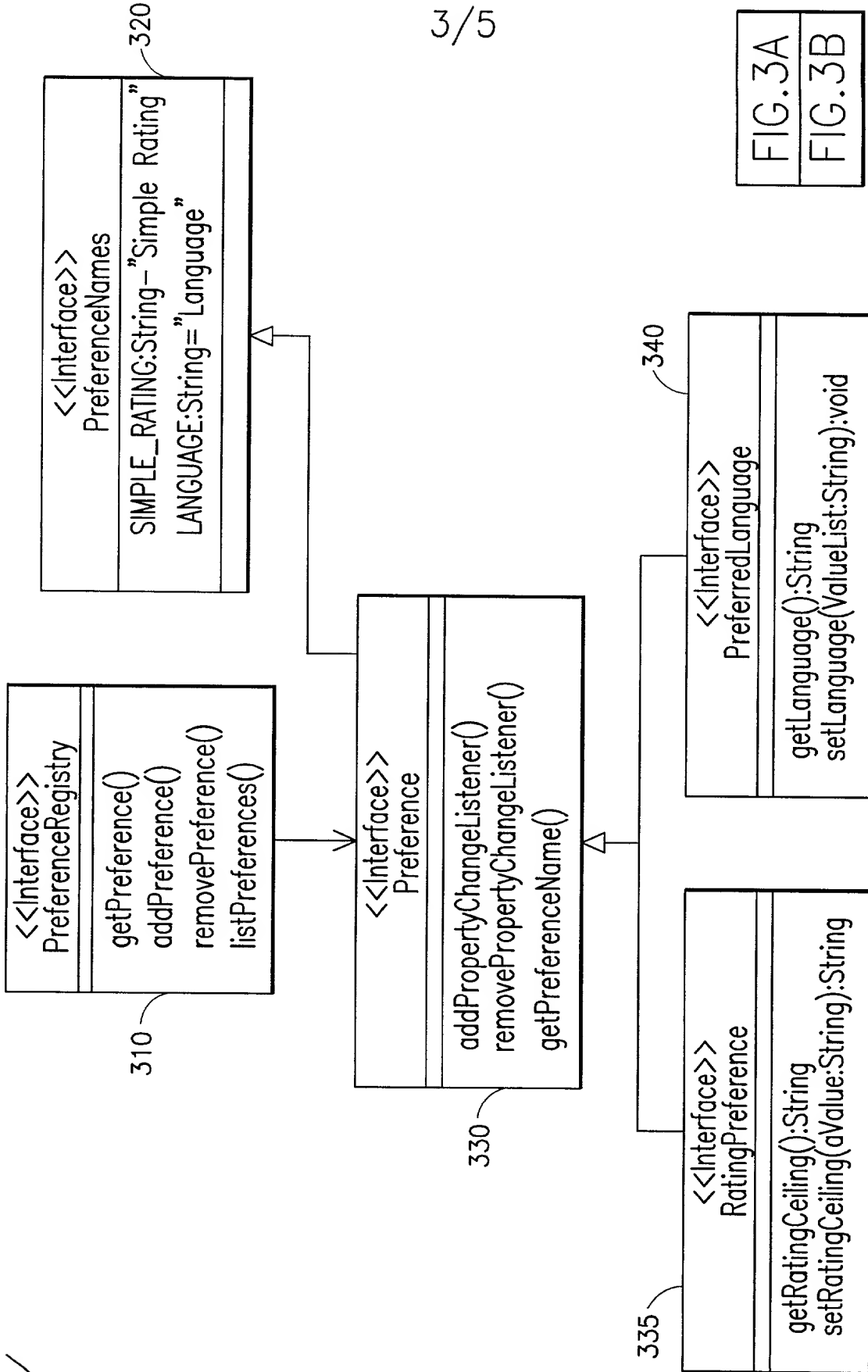


FIG. 3A

FIG. 3B

FIG. 3

FIG. 3A

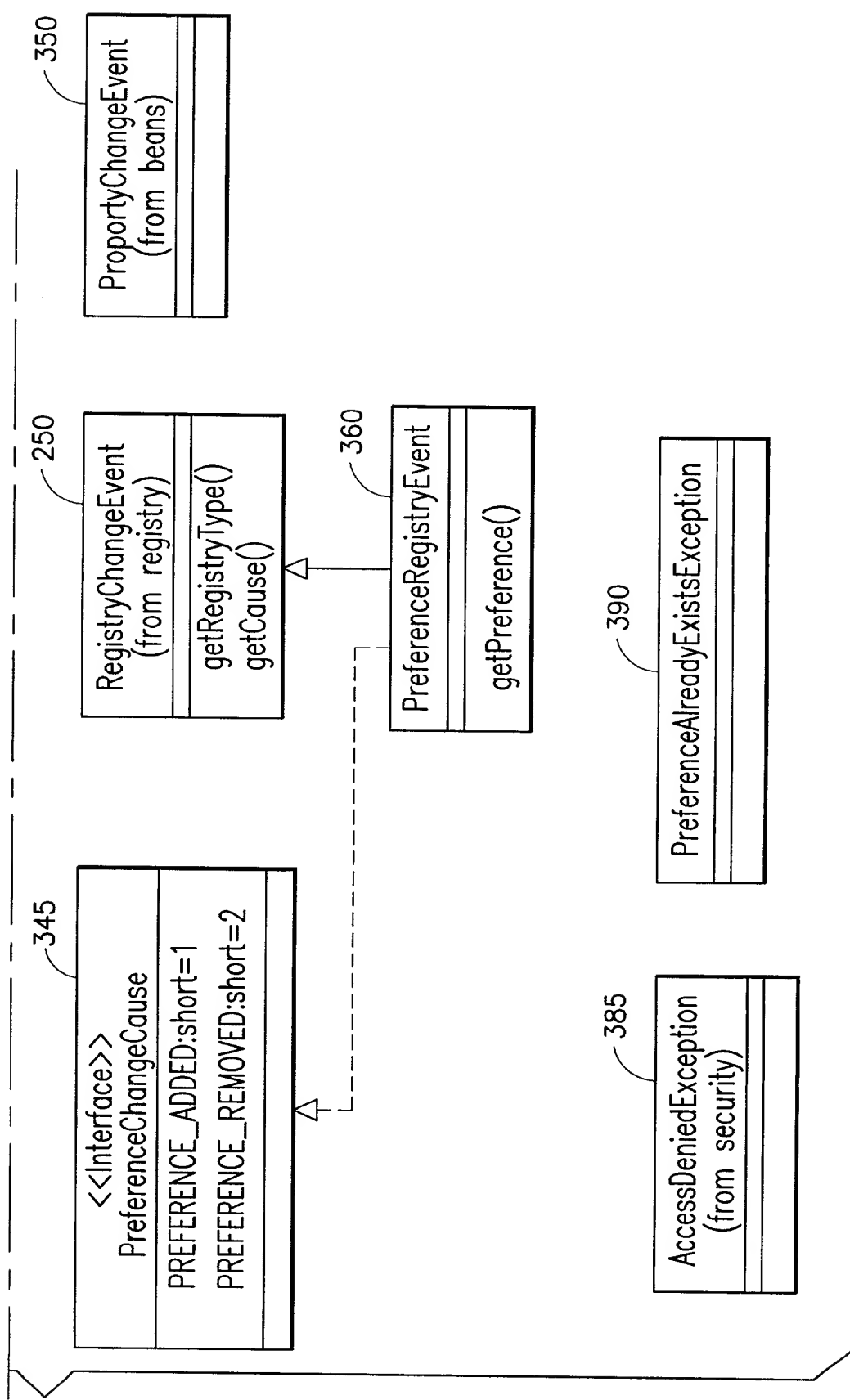


FIG.3B

5/5

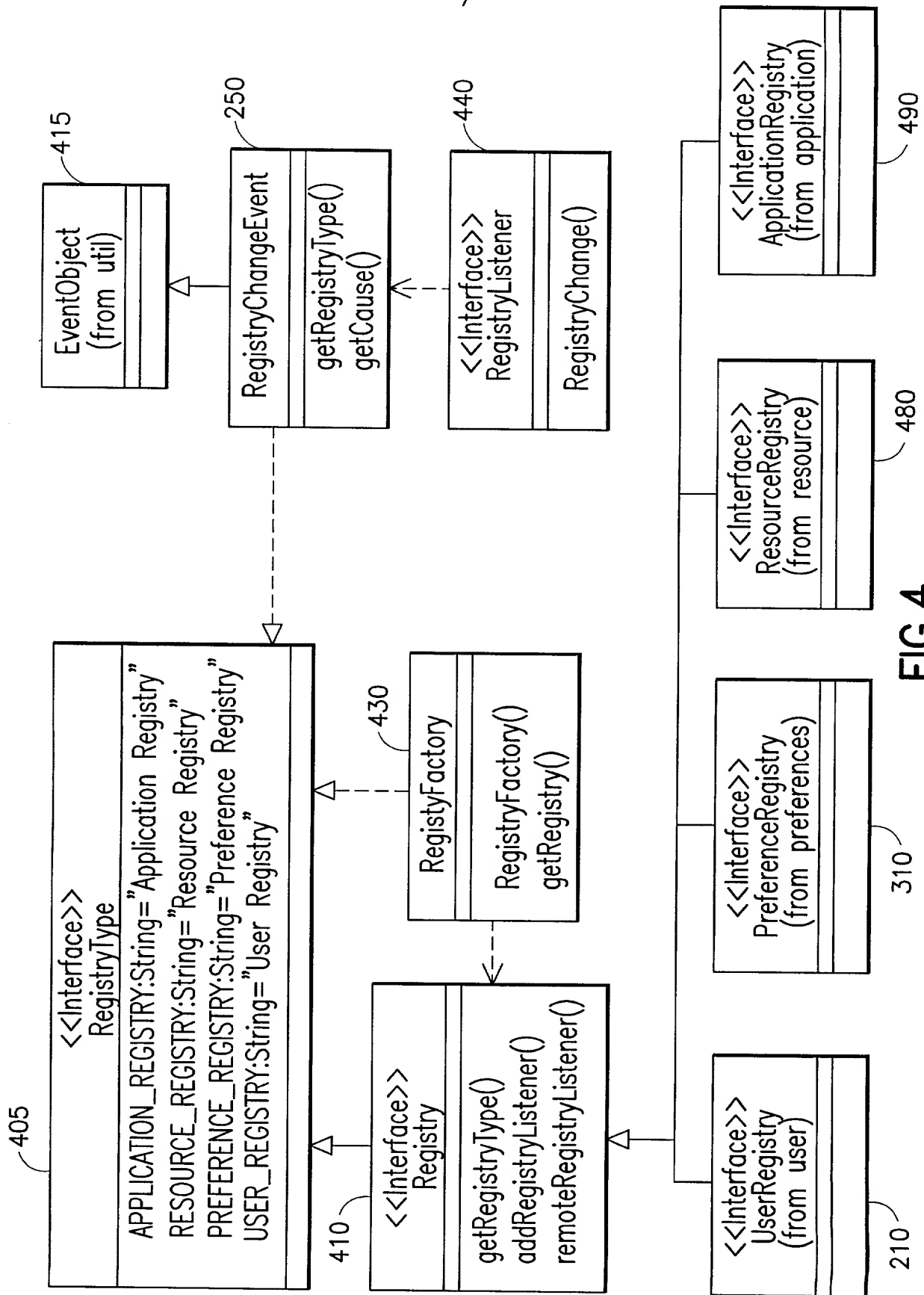


FIG. 4

DECLARATION, POWER OF ATTORNEY, AND PETITION

Attorney Docket No.: GIC-560

Page 1 of 2

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below next to my name.

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

DIGITAL TELEVISION RECEIVER WITH APPLICATION PROGRAMMING INTERFACE FOR USER MANAGEMENT

the specification of which is attached hereto unless the following box is checked:

[X] was filed on **October 7, 1999** as United States Application Number _____ or PCT International Application Number **PCT/US99/23346** and was amended on **March 22, 2001** (if applicable).

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claims, as amended by any amendment referred to above.

I acknowledge the duty to disclose to the U.S. Patent and Trademark Office all information known to be material to the patentability of this application in accordance with Title 37, Code of Federal Regulations, §1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, §119(a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate or 365(a) of any PCT international application which designated at least one country other than the United States of America, listed below and have also identified below any foreign application for patent or inventor's certificate or of any PCT international application having a filing date before that of the application on which priority is claimed:

(Number)	(Country)	Month/Day/Year Filed	Priority Claimed	
			[]	[]
			Yes	No
(Number)	(Country)	Month/Day/Year Filed	[]	[]
			Yes	No

I hereby claim the benefit under Title 35, United States Code, §119(e) of any United States provisional application(s) listed below.

60/107,949

November 12, 1998

(Application Number)

(Filing Date) - Month/Day/Year

(Application Number)

(Filing Date) - Month/Day/Year

I hereby claim the benefit under 35 U.S.C. 120 of any United States application(s), or 365(c) of any PCT international application designating the United States of America, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of 35 U.S.C. 112, I acknowledge the duty to disclose information which is material to patentability as defined in 37 C.F.R. 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application.

**U.S. Parent Application
or PCT Parent Number**

**Parent Filing Date
(MM/DD/YYYY)**

**Parent Patent Number
(if applicable)**

2

And I hereby appoint: Barry R. Lipsitz, Registration No. 28,637 and Douglas M. McAllister, Registration No. 37,886, all of the firm of Barry R. Lipsitz, Attorney at Law, 755 Main Street, Bldg. 8, Monroe, Connecticut 06468, Telephone (203) 459-0200, my attorneys with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected therewith.

Wherefore I pray that Letters Patent be granted to me for the invention or discovery described and claimed in the foregoing specification and claims, and I hereby subscribe my name to the foregoing specification and claims, declaration, power of attorney, and this petition.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

1-00

Full name of sole or first inventor: **Petr PETERKA**

Inventor's Signature *Petr Peterka* Date: 4/27/2001

Residence San Diego California CA Citizenship: USA
(City) (State or Foreign Country)

Post Office Address 5126 Caminito Vista Lujo San Diego, California 92130, U.S.A.
(Post Office Address) (City) (State & Zip Code/Country)

2

Full name of second joint inventor: **Branislav N. MEANDZIJA**

Inventor's Signature *Branislav N. Meandzija* Date: 6/17/2001

Residence Del Mar California CA Citizenship: USA
(City) (State or Foreign Country)

Post Office Address 716 Avocado Place Del Mar, California 92014, U.S.A.
(Post Office Address) (City) (State & Zip Code/Country)